RDF-based Modelling of Web Documents on Different Levels of Abstraction

Martin Milička and Radek Burget

Brno University of Technology,
Faculty of Information Technology,
Božetěchova 2,
612 00 Brno, Czech Republic

Abstract. The information presented in web documents can be modelled at different levels of abstraction. The paper introduces four levels of document description where three levels of visual information description in web documents correspond to different stages of document processing: page rendering model (box model description), page segmentation model and the semantic description of the document content. We also mention a possibility of a domain-oriented description of the individual document parts. We discuss the usual models used in the mentioned areas and we propose an RDF-based representation that allows representing the document simultaneously on different levels of abstraction.

1 Introduction

When processing the contents of web documents by a computer, it is necessary to create an appropriate model of the document. For HTML and XML documents, the most used one is the Document Object Model (DOM) that is suitable for modelling the internal structure of the document code. This model is commonly used in web browsers for computing the page layout. For other applications that should process the displayed document contents rather than the HTML code, other models have been proposed that describe the rendered page and the results of its visual segmentation [3]. During the document processing, further additional information must be added to the individual parts of the document contents such as a content class assigned during the text classification [2]. As we may see, for a single document, we need various models on different levels of abstraction.

Most document models such as DOM use a tree structure for representing the relationships among the individual parts of the document content. However, for modelling the visual perception of an HTML document, the tree structure is quite limiting because the visual information defines much more inter-element relationships. Thus, a general graph structure seems to be more suitable.

RDF is a well known implementation of a graph structure. Using elementary triplets connecting the *subject*, *predicate* and *object*, it allows creating complex graph structures of linked data. It also allows creating the connection between the low-level document elements and other models on higher levels. There exist standardized RDF data querying formalisms and it also gives the possibility to use the existing tools for reasoning.

In this paper, we define four levels of document description based on ontologies and RDF. We show the advantages of the ontology use for the document modelling. A special part is dedicated to low level document description where visual features create an important part. Thanks to the RDF description, we can easily access the represented data in the comparison with DOM model.

The paper is organized as follows. Section 2 provides the related research. Section 3 introduces the levels of possible document description and the design of two proposed low level ontologies. Finally, Section 4 draws the conclusions.

2 Related research

In the field of information retrieval and knowledge discovery, we can see an active research that works with visual features. For example, Liu et al. presented data record extraction based on the vision in [8]. Similarly, Penna et al. published an extraction of information based on the visual features in [11]. Such approaches usually have some connection to textual data to obtain better results.

A typical information retrieval approach works with a document model based on a tree structure. It is similar to Document Object Model (DOM). The limitation of such structure is the number of defined relationships between the elements.

Currently, the ontology-based approaches are being used for the multimedia description as well. For example, an ontological description is often used for the video scene description. Multimedia analysis ontology was presented in [6] where it is used for the assisted semantic video object detection. In [10], the authors are presenting ontology for dynamic video scene understanding. All works about the video description work with the MPEG-7 standard [9].

In the area of a general document description, Eriksson published the article [7] where he is storing RDF description in PDF files. This RDF contains the annotations of objects in PDF files. However, the visual information is not described in this model in detail. For the logical description of particular parts of a document, the *SALT ontology*[1] may be used. It defines the concepts such as the document content, headline, table, image, etc. This ontology was primary developed for LaTeX documents.

The above mentioned models and ontologies form the base for the document description based on visual features proposed in this paper.

3 Levels of document description

In general, a document model can be created at different levels of perception. We are proposing four levels of the document description in the figure 1. This figure can be split in two parts where the left part shows the individual steps of

[1] http://salt.semanticauthoring.org/ontologies/sdo

the document processing. The process starts with the page rendering and ends with obtaining the particular information relevant to specific domains. On the right, we can see the corresponding ontologies for the document description.

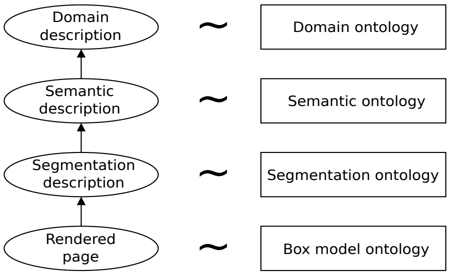

Fig. 1. Levels of document description

The visual features of the document contents play an important role especially in the first two levels of document processing.

Each level of the process has a strong link between the lower and the higher level. The lower level provides the input for the current level. The output of the current level provides the input for the higher level. The HTML source code and the visual features defined using CSS are the inputs of the rendering process that starts the whole processing. Each level has its characteristic output.

3.1 Box model ontology

This ontology describes the rendered document. The document rendering is based on the source data presented in the HTML document and the visual features defined using the Cascading Style Sheets.

In the CSS formatting model [1], a *Box* denotes a base element of the rendered document. It can be viewed as a rectangular area with certain position, width, height and its visual features in the rendered page. Similarly to the Document Object Model (DOM), the boxes are organized in a hierarchical structure.

In the figure 2 A), we can see the design of the *Box model ontology*. The class *Rectangle* creates a basic class of the defined ontology. It has its characteristic size, position and visual features. The selection of visual features follows the suggestions from article of Burget and Burgetová [2]. If the rectangle has a border, there is an object property *Border* defined for each side. Also every rectangle has an object property *belongsTo* that indicates the relationship to a specific document – Page.

The *Page* class represents the original document and contains the *sourceUrl* that denotes a unique URL of document data property. Another important class is *Box*. It is a subclass of the *Rectangle*. The *Box* class has some object properties that have a connection to its content. There is an object property *containsObject*

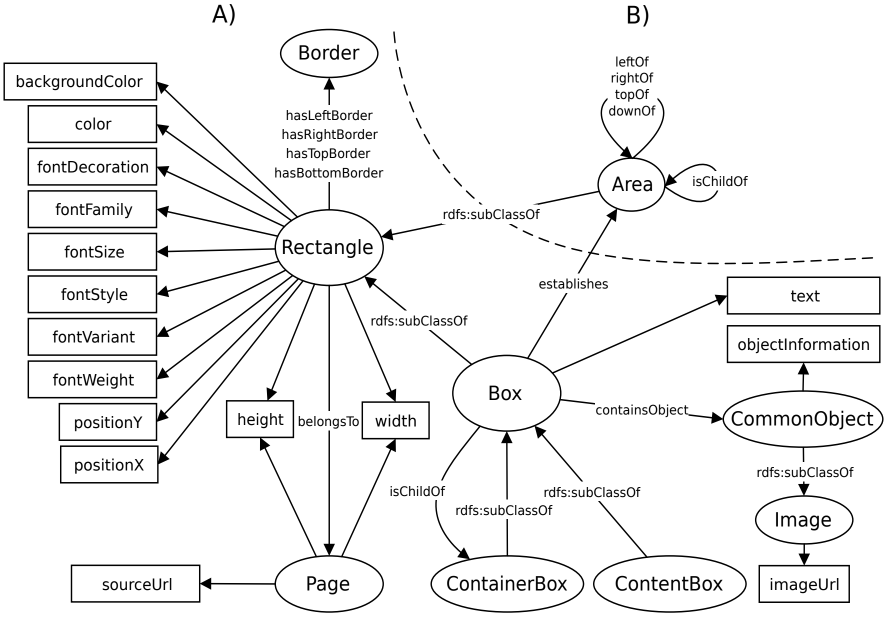

Fig. 2. A) Box model ontology B) Segmentation ontology

that contains a common information about the objects in the page. *Image* class is an example of the specific object.

The textual information is represented using the *hasText* data property.

The *Box* can be specialized into the *ContainerBox* or *ContentBox* classes. *ContainerBox* represents the possibility of box nesting. *ContentBox* represents Box that does not allow the box nesting; it contains the connections to final objects like images, common objects or textual information.

3.2 Segmentation ontology

The purpose of the page segmentation is to detect the individual visually distinguished segments of the document contents in the page. One of the most popular visual segmentation algorithms for web pages was introduced by Microsoft research group in [5]. This algorithm has many modifications. One of them also presented Burget and Rudolfová in [3].

The result of the segmentation is based on elements that have rectangular shape. The terminology of segmentation algorithms can be different for each author; however, the idea of the segmentation description is similar. For instance, Burget and Rudolfová are calling the basic element as *Area* [3] where Area is visually autonomous and enables Area nesting.

The Segmentation ontology extends the Box model ontology. The basic *Area* class is a specialization of the Rectangle class from Box model ontology. The ontology design is also presented on the figure 2.

The designed segmentation ontology uses the Area class for representing the visual areas detected during the page segmentation. The areas may be hierarchically nested and they may be also connected to the individual boxes from the Box model ontology.

The *Area* class has the object property *childOf* that has a cyclic dependency on the *Area* class. If the area is created by a particular box, the *establishes* object property represents the connection between the area and the box. However, some area may not correspond to any particular box since they may be created by a visual cluster of multiple boxes.

3.3 Semantic ontology

This phase tries to find or define the parts of content with a specific meaning. The description of a segmented document is an input of the semantic processing.

There can be two types of the semantic document description defined: a manual or an automatic annotation. In the manual annotation, the semantic description is done by user that assigns a meaning (a class) to specific areas. The manually assigned classes may be the used for creating an automatic classifier that is able to assign the classes to the new, previously unknown documents automatically, as proposed for example in [2]. Semantic ontology represents the classes assigned manually or automatically to the individual parts of the document.

The *SALT ontology* is an example of a semantic ontology for documents. It provides a logical description of particular parts of the document. It defines parts like *content*, *headline*, *table*, *image*, etc. This ontology was primary developed for LaTeXdocuments.

According to the visual features of areas that are defined in segmentation ontology, we can define rules for the detection of document content, menus, advertisements, headlines, etc. In general, this ontology has a close connection to the domain ontologies because it defines the semantic parts of document.

3.4 Domain ontology

The domain ontology describes a particular application domain of the published information. For the documents from the given domain, the individual parts of the document described using the previously mentioned rendering, segmentation and semantic ontologies may be assigned to some concepts of the domain ontology.

The examples of domain ontologies are the FOAF ontology, Event ontology, calendar ontology, etc. For instance, in the context of conferences we can define a conference program ontology. This ontology has classes like date, time, topic, description and authors.

For mapping the individual document parts to the particular concepts of the domain ontology, different approaches may be used such as an approximate tree mapping [4], visual feature classification [2] or Name Entity Recognition (NER).

4 Conclusion

In this paper, we have presented the general levels of the document description where each level is characterized its specific model. For the page rendering and segmentation levels, we have proposed the corresponding ontologies that may be used for describing a rendered and segmented page including the visual features.

This paper also creates an introduction into the key part of the document description which is *Semantic ontology*. It creates the document description based on the semantic rules.

Acknowledgement

This work was supported by the research programme MSM 0021630528, the BUT FIT grant FIT-S-11-2 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

References

1. Bos, B., Lie, H. W., Lilley, C., and Jacobs, I.: Cascading style sheets, level 2, CSS2 specification: The World Wide Web Consorcium, 1998
2. Burget, R., Burgetová, I.: Automatic annotation of online articles based on visual feature classification. International Journal of Intelligent Information and Database Systems, 5(4), 2011, pp. 338–360.
3. Burget, R., Rudolfová, I.: Web page element classification based on visual features. 1st Asian Conference on Intelligent Information and Databases Systems ACIIDS, 2009, pp. 67–72.
4. Burget, R.: Hierarchies in HTML Documents: Linking Text to Concepts. 15th International Workshop on Database and Expert Systems Applications, 2004, pp. 186–190
5. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Vips: A vision-based page segmentation algorithm. Microsoft Research, 2003.
6. Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.-K., Strintzis, M.G.: Knowledge-assisted semantic video object detection. Circuits and Systems for Video Technology, 15(10), 2005, pp. 1210–1224.
7. Eriksson, H.: The semantic-document approach to combining documents and ontologies. International Journal of Human-Computer Studies, 65(7), 2007, pp. 624–639.
8. Liu, W., Meng, X., Meng, W.: Vision-based web data records extraction. In: Proc. 9th International Workshop on the Web and Databases, 2006, pp. 20–25.
9. Martínez, M.J., Koenen, R., Pereira, F.: MPEG-7: The Genetic Multimedia Content Description Standard. Siemens Corporate Research, 2002.
10. Olszewska, J. I., Mccluskey, T. L.: Ontology-coupled active contours for dynamic video scene understanding. 15th IEEE International Conference on Intelligent Engineering Systems (INES), 2011, pp. 369–374.
11. Penna, G.D., Magazzeni, D., Orefice, S.: Visual extraction of information from web pages. Journal of Visual Languages and Computing, 21(1), 2010, pp. 23–32.